

# Release E - Debug

When following the instructions in the [Run in Docker](#), sometimes errors can happen. This page tries to provide some hints of how to solve some errors.

- [Log collection](#)
- [Did the A1 Policy Management Service start correctly?](#)
  - [Volume not mounted properly](#)

## Log collection

In order to collect logs from each service you can use the following command:

```
docker exec <container_name> cat <log_path>
```

Below there is a list with the container names followed by the log path, to replace the values in the command above. The container names correspond to the names used in the example Run page, to double check that you are using the same names in your environment, you could use `docker ps`.

**policy-agent-container**="/var/log/policy-agent/application.log"

**enrichment-service-container**="/var/log/enrichment-coordinator-service/application.log"

**nonrtric-control-panel**="/logs/nonrtric-controlpanel.log"

**nonrtric-gateway**="/var/log/nonrtric-gateway/application.log"

The below command can be used to check if there is any error in the logs, in this case for me `policy-agent-container`, but it could be checked for every other container.

```
docker exec policy-agent-container grep ERR /var/log/policy-agent/application.log
```

Warning can also be checked, using `WARN` instead of `ERR` in the command above.

If using SDNC controller, logs can be seen using the following command:

```
docker exec a1controller sh -c "tail -f /opt/opendaylight/data/log/karaf.log"
```

where `a1controller` is the name of the container and the path of the log is `/opt/opendaylight/data/log/karaf.log`

## Did the A1 Policy Management Service start correctly?

### Volume not mounted properly

```
2021-10-29 07:23:04.878 DEBUG 1 --- [ parallel-2] o.o.c.o.a.dmaap.DmaapMessageConsumer : InfiniteFlux.onRequest 1
2021-10-29 07:23:04.879 DEBUG 1 --- [ parallel-2] o.o.c.o.a.dmaap.DmaapMessageConsumer : fetchFromDmaap, no action DMAAP not configured
```

To be sure that the configuration file was mounted properly, execute the following command:

```
$ docker inspect -f '{{ json .Mounts }}' policy-agent-container | jq
```

The output below shows when the file was not properly mounted:

```
[
  {
    "Type": "volume",
    "Name": "f85d0d2c9c11dc9cd9cb20e6658711442fbd326ac8805ba2e5aa77d1c80af785",
    "Source": "/var/lib/docker/volumes/f85d0d2c9c11dc9cd9cb20e6658711442fbd326ac8805ba2e5aa77d1c80af785/_data",
    "Destination": "/opt/app/policy-agent/data/application_configuration.json",
    "Driver": "local",
    "Mode": "",
    "RW": true,
    "Propagation": ""
  }
]
```

When the file is mount correctly the output must look like this:

```
$ docker inspect -f '{{ json .Mounts }}' policy-agent-container | jq
```

```
[
  {
    "Type": "bind",
    "Source": "/home/ychacon/repo/onap/ccsdk/oran/a1-policy-management/config/application_configuration.json",
    "Destination": "/opt/app/policy-agent/data/application_configuration.json",
    "Mode": "",
    "RW": true,
    "Propagation": "rprivate"
  }
]
```

where the source should specify the absolute path of the file.

Remember that when you start the container with the command below, you need to replace "<absolute-path-to-file>" with the the path to the created configuration file in the command.

```
docker run --rm -v <absolute-path-to-file>/application_configuration.json:/opt/app/policy-agent/data
/application_configuration.json -p 8081:8081 -p 8433:8433 --network=nonrtric-docker-net --name=policy-agent-
container nexus3.o-ran-sc.org:10002/o-ran-sc/nonrtric-policy-agent:2.3.0
```